55185C

IAF.U.4
Collaboration Between NASA Centers of Excellence on Autonomous System Software Development

By
Charles H. Goodrich, Dynacs Engineering, Kennedy Space Center, FL
and
William E. Larson, NASA – Kennedy Space Center, FL

ABSTRACT

Software for space systems flight operations has its roots in the early days of the space program when computer systems were incapable of supporting highly complex and flexible control logic. Control systems relied on fast data acquisition and supervisory control from a roomful of systems engineers on the ground. Even though computer hardware and software has become many orders of magnitude more capable, space systems have largely adhered to this original paradigm

In an effort to break this mold, Kennedy Space Center (KSC) has invested in the development of model-based diagnosis and control applications for ten years having broad experience in both ground and spacecraft systems and software. KSC has now partnered with Ames Research Center (ARC), NASA's Center of Excellence in Information Technology, to create a new paradigm for the control of dynamic space systems. ARC has developed model-based diagnosis and intelligent planning software that enables spacecraft to handle most routine problems automatically and allocate resources in a flexible way to realize mission objectives. ARC demonstrated the utility of onboard diagnosis and planning with an experiment aboard Deep Space 1 in 1999.

This paper highlights the software control system collaboration between KSC and ARC. KSC has developed a Mars In-situ Resource Utilization testbed based on the Reverse Water Gas Shift (RWGS) reaction. This plant, built in KSC's Applied Chemistry Laboratory, is capable of producing the large amount of Oxygen that would be needed to support a Human Mars Mission. KSC and ARC are cooperating to develop an autonomous, fault-tolerant control system for RWGS to meet the need for autonomy on deep space missions. The paper will also describe how the new system software paradigm will be applied to Vehicle Health Monitoring, tested on the new X vehicles and integrated into future launch processing systems.

INTRODUCTION

As we enter new millennium there is little doubt that mankind's next great adventure, a human landing on Mars, will occur sometime this century. No other human space flight activity generates as much excitement in the minds of our engineers, scientists and the public. While we may never again see the confluence of events that led to the national mandate of the Apollo era, a determined set of engineers and scientists inside and outside of NASA have begun to develop the building blocks that will one day take us to the Red Planet.

These building blocks are essential to the realization of human flight to Mars. The paradigms that have existed human space flight since it's inception will no longer work for a Mars mission. First of all, the duration of these missions will exceed anything ever attempted. Depending on the mission class chosen, opposition or conjunction, the duration of the mission will be between 640 & 910 days.[1] While the Russians and USA have kept space stations operating in low earth orbit for that period of time, or longer. It was accomplished with continuous re-supply missions that ferried consumables and spare parts. A human Mars mission will have to be totally self sufficient for the duration of it's mission. Secondly, the distance traveled from earth will be so great that communications delays in the vicinity of Mars will vary from 20 to 40 minutes, depending on the orbital position of the planets.

To address these problems, work is underway at a number of NASA Field Centers to develop the technologies that will allow self-sufficiency. One area that has drawn great interest, because of it high payback in reduced mission mass, is the utilization of in-situ resources. Utilizing the atmosphere of Mars it is possible to produce all of the oxygen, fuel, buffer gasses and water needed for a long duration stay on the surface and the return trip home. This approach would significantly reduce the size of the launch vehicle needed to start the mission and reduce the mass of the

Mars Lander as well.*

To make these technologies practical, it is essential that they be robust, have built in redundancies, and a flexible, low maintenance and autonomous control system. As we have learned from the Russian and USA space station programs, a significant amount of crew time ends up being spent on maintenance activities. This must be reduced so that the crew can spend more time performing scientific and exploration activities rather than acting as a "handyman". The communications delays mentioned above preclude the use of ground controllers to control and operate these systems so new autonomous control technologies seem to be the only viable answer to minimize crew workload yet allow the use of critical in-situ resource utilization systems. Before launching into a discussion of this new technology let's take a look at the history of flight operations software.

ROOTS OF FLIGHT OPERATIONS SOFTWARE

In the earliest days of the space program, software for space flight operations consisted of fast data acquisition or telemetry systems coupled with supervisory control by human operators. Relatively little automation was needed for these systems. Normal procedure was to gather a room full of engineers – experts in all the different spacecraft subsystems – and task this group to monitor data during the countdown, activate equipment in a pre-planned sequence, and correct for any problems detected. In the late 50's, the technology did not exist to place a significant share of the control responsibility in the systems software. Also, analog controllers used in the spacecraft and on ground processing systems were crude by today's standards. These factors required constant supervision by engineers to insure consistent, safe operation.

This team of experts was an important reason for the extraordinary success of the early Space Program. Ground operations for current space launches continue to use the large-team paradigm. While it is still an effective way of doing business, it also contributes to the non-routine nature of space flight and is a major reason for the inefficiency and high cost of Space Systems.

WHY MORE AUTONOMY IS NEEDED

The complexity of the launch processing system has increased along with that of the spacecraft itself. The Space Shuttle has on-board computers that sequence countdown operations in the last two minutes before launch. During this critical period, the shuttle computers talk to the Ground Launch Sequencer that is responsible for controlling ground support equipment. Human operators supervise the system and can override the automatic sequence if a problem is detected. Near the end of a Shuttle countdown in 1986, a limit switch falsely indicated that cryogenic propellant was sill flowing into the Shuttle External Tank after a command to close the fill/drain line. The launch team correctly determined that the switch had frozen because of exposure to super-cold propellant and that the valve in question was actually closed. Other sensors confirmed that propellant flow had stopped, so the launch team decided to override the faulty indication and continue the countdown. Unbeknownst to the engineers, the shuttle's on-board computer was programmed to *believe* the faulty switch and to keep another critical valve open to prevent its being damaged. This open valve actually allowed a large quantity of propellant to flow *out* of the shuttle's external tank. So much propellant drained from the tank that the Shuttle would not have reached orbit had it been launched. The problem was discovered after the flight was cancelled for another problem.

Communications delays also hamper operations under the traditional paradigm. In 1997, a software bug caused the Mars Pathfinder computer to reset itself four times in the first few days after landing. Mission engineers eventually found and fixed the problem, but their efforts were hindered by communications delays between Earth and Mars that effectively enabled only one uplink per day to the Lander.

These examples highlight problems with the traditional paradigm for flight operations software. Among other things, the large-team approach is inherently costly and labor intensive, a single faulty measurement can compromise sophisticated software systems, and remote troubleshooting can decrease the time available for scientific exploration. Clearly, more autonomous software capable of unattended operation and adaptive decision-making is needed. The systems software must take on added responsibility for the identification and resolution of problems. This software must have the same systems knowledge that today's ground controllers possess. It must be able to reason about system degradation and detect faulty sensor readings that indicate a problem where none truly exists like the limit switch described above.

* A more detailed discussion of in-situ resource utilization technologies can be found in the paper "Technology Development for Human Exploration of Mars in the New Millennium, IAA-1-IAA-13.3.07

AUTONOMOUS SOFTWARE IS MADE POSSIBLE BY MORE CAPABLE HARDWARE

Today's mission control hardware and software is considerably more capable and sophisticated than technology that was deployed only a decade ago. Computer hardware is much more reliable than early generation equipment. High availability and lower maintenance costs are much in demand by business and industrial users. NASA is not the only customer that requires 24/7 operation with 99.99% up time. The market has produced faster, better AND cheaper computers and peripherals. Imbedded controllers are much more powerful and user-friendly. They are used extensively in harsh industrial environments; and as a result are much more reliable.

The past decades have also witnessed significant improvements in software methodology. Mission critical software is now capable of more complex control with fewer bugs. Although many new systems employ higher autonomy, software sophistication has not kept up with improvements in hardware. Most control systems still rely on human supervision. Typically, only routine operations are fully automated; a few error cases may be explicitly handled, but really complex or unusual situations are almost always deferred to the operator. Some automation systems cannot even *detect* many categories of errors; management still relies on carefully trained and experienced operators to recognize and avoid expensive breakdowns.

AUTONOMOUS SOFTWARE DEVELOPMENT AT KSC

To address some of these issues and advance the state of the art, Kennedy Space Center initiated Artificial Intelligence (AI) research in Model-Based Reasoning (MBR) beginning in 1983. Over the next twelve years, applications were developed for environmental control, propellant loading, single point failure analysis and, thermal control. Each of these applications demonstrated Model-Based control and diagnosis. By using a model of the physical system, MBR software was able to detect faults that were not pre-programmed in a symptom/cause style database. Instead, the software used a mathematical model of the physical system like a financial analyst would use a spreadsheet to test "what-if?" scenarios for failure of components in the context of current system operation. This method allowed many defective components to be detected without exhaustively cataloging system-wide symptoms under a range of operating conditions. In addition, the system was able to keep on detecting new faults after initially recognizing a defective component by simulating its "broken" behavior in subsequent diagnoses. In addition to diagnosis, these systems were able to use the math model to calculate control actions such as temperature control and redundancy management. These control actions were achieved successfully even in the presence of failed components. Mission control software had thus become more capable *and* more fault-tolerant.

CASSINI SIMULATOR AND DS1 DEMONSTRATION BY JPL AND AMES RESEARCH CENTER (ARC)

In 1995 NASA began the New Millennium Program (NMP) to reduce the cost of space-exploration while shortening the schedule for spacecraft deployment. The goal of NMP is to validate new technologies including autonomous control for 21st-century science missions. The vision is to be able to "fire and forget" a whole series of missions that will go about their business of exploring, contacting home only when they find something of scientific interest or need help. Each spacecraft would manage its own travel, malfunctions, and much of the science. KSC had employed MBR and autonomy only for ground-support applications, but NMP was intended for spacecraft control. When NMP was initiated in 1995, the AI groups at ARC and JPL met with spacecraft-engineering experts to begin designing software architecture for autonomous operation of NMP spacecraft. After a five-month development effort the resulting AI system, Remote Agent (RA), was tested on JPL's Cassini simulator. RA successfully inserted a simulated spacecraft into orbit around Saturn. During the simulated mission, RA had to trade off science and engineering priorities and achieve mission goals in the face of numerous computer-generated hardware failures. As a result of the simulated mission success, RA was selected and successfully deployed as an experiment on the first NMP flight, Deep Space One (DS1) in 1998.

IN-SITU RESOURCE UTILIZATION (ISRU)

In-Situ Resource Utilization has become a key component of NASA's plans for Mars exploration.
Major cost savings are possible when propellants and other consumables are manufactured on Mars instead of being imported from earth. The reverse water gas shift (RWGS) process is one of the technologies being developed to address this need. RWGS is a chemical reaction that produces oxygen (O_2) from the atmosphere of Mars that is mostly carbon dioxide (CO_2).

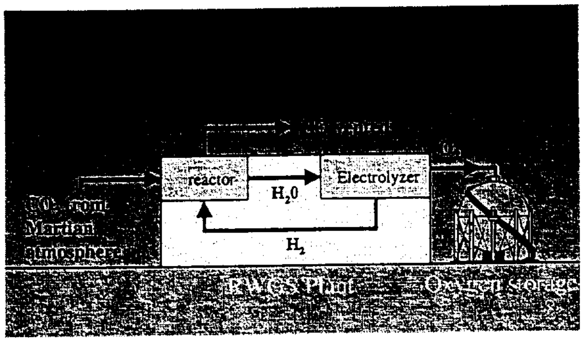

Figure 1 – RWGS Plant on Mars

The RWGS reaction occurs when CO_2 is combined with hydrogen (H_2). The water produced is electrolyzed the O_2 from electrolysis is stored, and the H_2 is recirculated into the input stream as shown in Figure 1. Since all the H_2 is reused, only a small amount needs to be imported from earth. The net result of the RWGS plant is to produce as much O_2 as needed for propellant or life support using only CO_2 from the Martian atmosphere as a raw material.

As a byproduct of the reaction, The RWGS reactor vents carbon monoxide (CO). If the molar ratio[2] of feed gases is not exactly 1:1, excess H_2 or CO_2 will also vent. The control challenge is to supply CO_2 and H_2 to the reactor in exactly the right amounts to maximize production and avoid wasting either gas - particularly H_2 that is relatively scarce on Mars. Among other things, the plant control system must monitor the vent stream, detect if either feed gas is in excess and adjust flows to continue efficient operation. RWGS must operate for several years on Mars without human intervention, so its control system must be autonomous and highly reliable. One of the most common fault modes for process equipment in harsh environments is instrumentation failure. The control system must supply missing instrumentation data and correctly compute real-value control actions.

COLLABORATION BETWEEN ARC AND KSC TO ADDRESS COMPLEXITY OF ISRU

All of the control challenges of RWGS are well suited for systems autonomy. KSC has extensive experience in propellant loading and ground operations – a good background for developing ISRU. In addition, KSC has knowledge in autonomy and process control software that are needed for the control system. In 1997 ARC and KSC began collaborating on ISRU. ARC, as the center of excellence in Information Technology, supplied and supported the Livingstone MBR software for ISRU. KSC developed and tested autonomy applications with the assistance of ARC personnel. NASA managers decided that KSC would build and test a RWGS testbed after evaluating several candidate ISRU technologies. Since other NASA centers were already at work on alternate ISRU processes, RWGS was a good choice for KSC. The project had three goals:

- Characterize the process and collect operating data
- Improve efficiency and technology readiness level (TRL)
- Demonstrate autonomous control of ISRU

Progress has been made in all three of these important areas. Operating data on RWGS indicates that the process can be a cost-effective source of O_2 for planetary missions; RWGS energy efficiency and critical control parameters have been measured; and unique requirements of ISRU process control have been identifed that will enhance the TRL of ARC Livingstone software and expand its capability to control new types of processes and systems.

System Variables Key To Autonomy Success

ARC had applied discrete abstractions of continuous system variables in all of their MBR applications prior to RWGS. The collaborative effort between KSC and ARC was to answer an important question: "Would this approach work for ISRU?" Following is a brief discussion of the issues involved in discrete vs. continuous models of spacecraft control systems.

Continuous vs. Discrete Models

In order to use MBR algorithms such as Livingstone, a model or formal description of the physical system is needed. A *continuous* model includes variables that may take on an infinite number of values and continuous functions that can compute them. Model values are normally expressed as real numbers such as 56.8 psia or 425.2 °C. Change is modeled as the derivative of one or more variables over time. For example, a model of heating and cooling of a spacecraft might include continuous variables that represent the energy radiated from the spacecraft into space, the energy output of its heaters, and a set of differential equations that model thermal conductivity through the spacecraft structure.

The physical behavior of such systems is continuous, but it is often possible to capture the features of the system that are relevant to diagnosis and control in a *discrete* model of the system. Consider the following example.

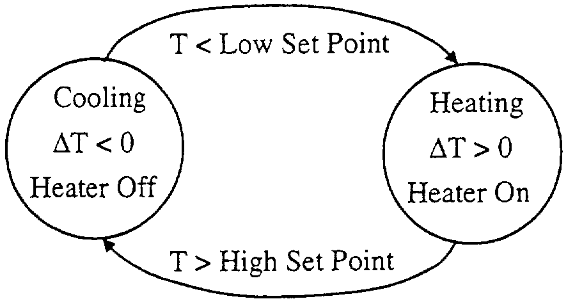

Figure 2 – Spacecraft Temperature Model

In this model of a spacecraft thermostat, an infinite number of continuous behaviors have been abstracted into a finite number of discrete categories. When the temperature level drops below a set point, the system enters the discrete heating state where the heater must be on and the temperature must be rising ($\Delta T>0$). Any continuous behavior that fits this discrete description is normal behavior, and any that does not indicates a failure. Discrete AI techniques such as Livingston can be applied successfully if an abstraction such as the one in Figure 2 provides sufficient information for the task at hand. The models used for MBR on the Cassini simulator and on DS1 spacecraft were also discrete. Figure 3 illustrates the propulsion system used in Cassini. The purpose of the system is to provide thrust to insert the spacecraft into orbit around Saturn. This is accomplished by combining fuel and oxidizer in an engine for a specified amount of time. A tank contains helium under high pressure. When the appropriate valves are opened, helium pressurizes the oxidizer and fuel tanks. This forces oxidizer and fuel into the engine where it is ignited to produce thrust. When sufficient thrust is achieved, the valves are closed.

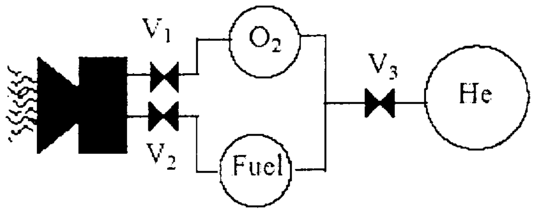

Figure 3 – Simplified Cassini Propulsion System

While the Cassini application appears to address continuous variables such as pressure and flow, the propulsion system model is something of a special case. Since it consists of a pressurized tank at one end and vacuum at the other, the control flow and pressure gradient are always in one direction. There are no closed loops or mixing of flows as in RWGS. The diagnosis problem was only concerned with abrupt failures such as stuck valves, and did not attempt to capture leaks, flow reversions, or more subtle anomalies. No diagnosis required observing the system over time. Diagnosis used only context-free discrete observations (e.g., flow, no flow). Similarly, the control problem was defined in terms of discrete actions such as opening and closing valves. There was no need to estimate and adjust continuous parameters of the system such as flow rates.

In applications like Cassini and DS1, a discrete, qualitative model is often effective because the model is easy to construct and the algorithms for reasoning about it are very efficient. In contrast, RWGS and similar systems involve branching or multi-directional fluid flows, electrical currents or similar quantities, and capacitance, such as storage tanks or buffers, that change over time. ARC and KSC found it difficult to produce a discrete abstraction of RWGS that was both consistent with respect to diagnosis and relevant with respect to control. In the following section we provide some intuitions from the RWGS domain on where the trouble arises and how KSC and ARC are working to overcome the difficulties.

Problems encountered with initial "discrete" RWGS model

In order to produce a qualitative model of a continuous system, we must discretize the variables that model the system's behavior. In the thermostat example, we discretized a continuous model of the temperature change over time into a discrete model specifying whether the derivative of the temperature was positive or negative. In order to develop a qualitative model of the RWGS system, we initially characterized continuous variables such as H_2 flow rate as "low, expected or high". This allowed us to start writing discrete constraints, for example relating the qualitative flow rate of H_2 and CO_2 to the qualitative rate of H_2O production. Intuitively, if the H_2 input to the reaction is lower than expected, then the H_2O will be low as well.

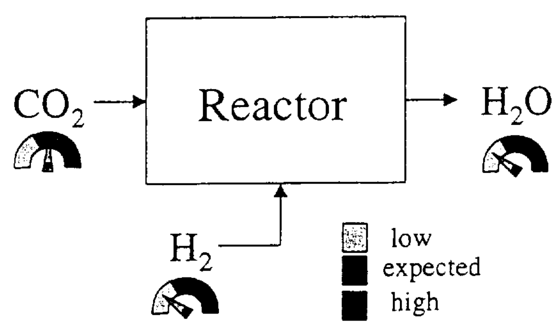

Figure 4 – RWGS discrete constraints

It soon became clear this qualitative abstraction of the system was inadequate. Often, a "low" value was the expected value. For example, if we intentionally turn a valve off in a redundant, unused branch of the system, then the expected flow rate in the branch is zero. The sensed value zero thus corresponds the qualitative value "expected". However, if the H_2O output of the reactor is zero, the qualitative value is "low".

There are other cases where it is desirable to reduce the output of reactor by simultaneously cutting back on the H2 and CO2 flows. We do not need to diagnose a failure to explain the reduced production rate under these circumstances. There were circumstances in RWGS that required comparisons between quantitative values; these relationships were almost impossible to model qualitatively. In particular, the reactor function in Figure 4 compares two real-valued parameters, H_2 flow and CO_2 flow. The smaller of the two flows – also called the limiting reagent – determined the resulting H_2O production. We could not adequately capture this relationship - i.e. how to determine the "lower" of two "lows" - within the discrete model.

Success with the RWGS simulator on test cases

The discrete model developed by KSC and ARC was subjected to testing with an Excel simulation of the RWGS testbed. Results of these tests highlighted some of the strengths of the discrete model. KSC and ARC developed an interface between the Livingstone software and Microsoft Excel spreadsheets. The interface uses a Livingstone Model to write an Excel macro. The macro generates a spreadsheet that updates Livingstone whenever a command or measurement in the simulation changes. Excel is used by many industrial and manufacturing organizations to model behavior of physical and chemical processes. The Excel/Livingstone link allows rapid prototyping of Livingstone models and testing against simulated conditions in a spacecraft or factory. KSC developed a detailed Excel model of the RWGS process including all the commands and measurements in the testbed. Several failure scenarios were tested with the simulator including level sensor failures in the water tanks and heater failures in the RWGS reactor. These tests demonstrated the utility of MBR as a high-reliability control system for ISPP operations on Mars.

RWGS Highlights Benefits of MBR

Operation of the RWGS test bed at KSC has shown that an MBR control system will yield important advantages for autonomous operation on Mars. A math model of the chemical process helps compute control actions and diagnose instrumentation failures over the long operating life that is required for the system. For example consider feed gas control in RWGS: As shown in Figure 1, the system vents CO; but if the feed rates are not exactly equal, it will also vent any excess H_2 or CO_2 left over from the reaction. A mass spectrometer attached to the vent stream could be used to detect out-of-balance feed; but cost, complexity and weight considerations make routine use of a mass spectrometer impractical for control. Data collected from the RWGS testbed at KSC show that it is possible to obtain composition data from the RWGS vent flowmeter by using related measurements and model-based techniques as follows:

Mass flow meters are specifically calibrated only for the gas that they are intended to measure. Nevertheless, it is possible to compute the flow of another gas by using a "K" factor to correct the apparent flow for thermal characteristics of the different gas. Since the K factor of CO_2 is large and factors for H_2 and CO are smaller, it is relatively easy to spot excess CO_2 in the vent. Excess H_2 can be detected by comparing the vent flow to feed rates of H_2 and CO_2 and the rate of H_2O production reported by other measurements. These advantages were disclosed in a New Technology Report in March of 2001.[3]

New Tools Under Development for Working With Continuous Models

KSC and ARC have investigated new tools and techniques for dealing with continuous parameters in MBR systems. Promising approaches have been described in a paper published at the 6th International Symposium on Artificial Intelligence, Robotics and Automation in Space.[4] Continuous parameters that are part of an MBR system can be computed using a constraint network[5]. Figure 5 illustrates such a network for converting between temperature scales. By inverting the calculation parameters it is possible either to calculate Celsius temperatures from Fahrenheit or Fahrenheit from Celsius.

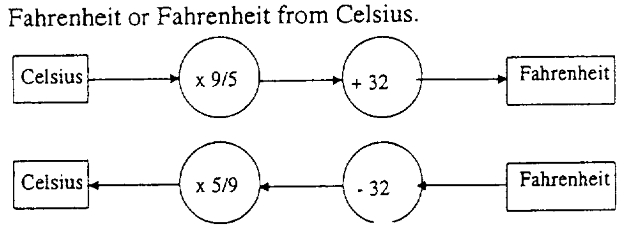

Figure 5 – Constraint network for temperature conversion

A constraint network functions like a spreadsheet except that calculations are multi-directional and the network is set up to make it easy to locate the source of conflicting constraints. In a manner similar to the temperature calculations above, commands and measurements from RWGS can be viewed as constraints in a network of calculations that are part of the math model for the process. Part of this network is shown in Figure 6. The calculations enclosed in circles can be inverted so that a selected parameter can be calculated by reference to the others; for example, electrolyzer current can be computed from H_2 flow or H_2 flow can be computed from electrolyzer current. Likewise, as described above, the composition of the vent gas can be inferred from the vent flow rate along with other measurements.

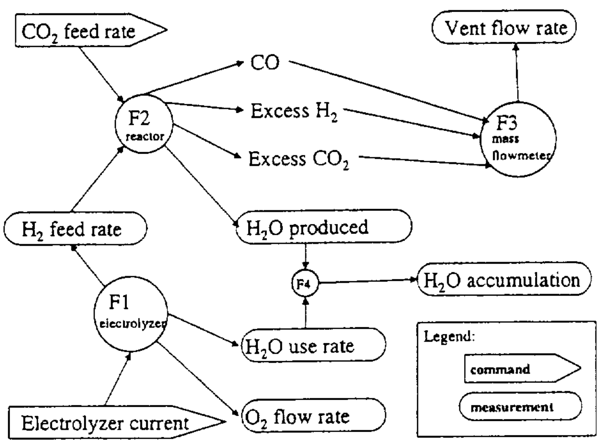

Figure 6 – Constraint network for RWGS

By the use of constraints, faulty measurements or other failed parameters can be identified and desired control actions easily computed.

Added Benefits of RWGS Data Collection

Characterized operation of RWGS

The data collected from the RWGS testbed includes measurements of power consumption and O_2 mass referenced to international standards. The testbed is scaled to produce enough O_2 to supply a typical Mars sample return mission, and KSC designers planned the dimensions and weight of the prototype so that it would fit the allowable size/mass envelope for a representative Mars lander experiment. These data will supply NASA with valuable references for use of RWGS on future ISRU missions.

Data for testing Bayesian Networks in MBR

RWGS is also the subject of a study sponsored by NASA/ARC at Stanford University. Researchers there are using Bayesian networks to infer the probability of events of interest in RWGS including diagnosing faults. Computational techniques are under development to answer questions to queries about system health given the probability of certain components being in a given state and observations of system parameters. This field is an active area of interest in AI research.

Application to Vehicle Health Monitoring.(VHM)

Researchers at ARC have developed a Livingstone model that will test MBR control on the X-37 project. X-37 is an advanced technology flight demonstrator, which will demonstrate advanced airframe, avionics and operations technologies that can support various launch vehicle and spacecraft designs. There is also considerable interest in MBR for Vehicle Health Monitoring (VHM) – a concept where autonomous systems test operation of critical systems while they are in flight. Reference to a model of the vehicle systems of interest is a powerful method to improve the performance of VHM and make it more versatile and fault-tolerant as described in the forgoing paragraphs.

CONCLUDING REMARKS

We have shown that MBR is a powerful technique for control of spacecraft and systems in an era of exploration that is demanding fault-tolerant software and autonomous systems. We have demonstrated the value of the ARC/KSC team approach in developing and improving autonomy software. This project highlighted KSC skills in developing potential flight-qualified experiments and controls. More powerful hardware and software is making possible more capable and effective space exploration missions in the 21st Century.

ACKNOWLEDGMENTS

The authors are indebted to the RWGS development team at Kennedy Space Center, FL and the Model-based Autonomy group at Ames Research Center whose work made this publication possible including Peter Engrand, Sharon Feagan, Mike Haddad, Douglas Hammond, Tim Hodge, Curtis Ihlefeld, Dale Lueck, Carl Mattson, Bob McLaughlin, Pamela Mullenix, Clyde Parrish, Guy Smith, John Taylor, Susan Waterman, Dan Clancy, Bill Millar, Chris Plaunt, Adam Sweet, Anupa Bajwa, Charles Pecheur, Benoit Hudson and Shirley Pepke

---

[1] Dr. Robert Zubrin, The Case for Mars, page 79, Touchstone, 1997.

[2] Stoichiometry deals with calculations about the masses of reactants and products involved in a chemical reaction. When two reactants are supplied in the correct "molar ratio" both are completely converted into products with no excess of either left over.

[3] NASA New Technology Report KSC12283.

[4] C. Goodrich and J. Kurien, Continuous Measurements and Quantitative Constraints - Challenge Problems for Discrete Modeling Techniques, *iSAIRAS* 2001.

[5] Harold Abelson and Gerald Jay Sussman, *Structure and Interpretation of Computer Programs*, pages 286-295, MIT Press, 1996.